

Windows C 语言网络编程快速入门

► 撰文 / 魏云华

C 语言的学习,一般的方式是:先学 C,然后是 C++,最好还要有汇编语言和微机原理基础,然后才是 Visual C++。这样的方式,对学习者来说,要花费很多时间和精力。而在学校教学中,也就没有时间深入学习 Windows 编程的实用技术了。

其实,具有了 C 语言基础后,再有一些基本的 C++ 类的概念,就可以直接学习 Windows C 编程了。

一、走近 Windows C 语言

很多语言都把显示一个“Hello,World!”做为第一个入门程序,C 语言的第一个程序是这样的:

```
#include<stdio.h>
main()
{
    printf("Hello,world!");
}
```

如果把 main 函数写成带参数的 main 函数,应该是:

```
#include<stdio.h>
main(int argc,char *argv[])
{
    printf("Hello,world!");
}
```

Windows C 的第一个程序和这个程序在形式和原理上都是一致的,只是有两点不同:

1. 主函数接收的形参不只是命令行中的字符串的个数和字符串的首地址。

2. C 语言的很多函数在 Windows C 中都可以继续使用,但象 printf() 屏幕显示等函数就不能继续使用了。因为 Windows 是多任务操作系统,屏幕已不再为某一个应用程序所独有,Windows C 应用程序要显示字符串,需要使用 Windows 提供的 API 函数,开自己的窗口。

下面是一个最简单的,显示“Hello,World!”的

Windows C 程序:

```
#include<windows.h>
APIENTRY winMain(HINSTANCE hInstance,HINSTANCE
hPrevInstance, LPSTR lpCmdLine,int nCmdShow)
{
    MessageBox(NULL,"Hello,World!","第一个Windows C程序",
    MB_OK|MB_ICONASTERISK);
}
```

主函数的形参有四个:

- hInstance: 接收程序运行时当前实例的句柄;
- hPrevInstance: 前一个实例的句柄;
- lpCmdLine: 程序命令行指针;
- nCmdShow: 一个用来指定窗口显示方式的整数。

这几个参数的使用我们会在深入的学习中介绍的。

显示“Hello,Word!”字符串,我们使用了一个 MessageBox 函数,这个函数会在屏幕上显示一个对话框,它的原型是:

```
int MessageBox(HWND hWnd,LPCTSTR lpText,
LPCTSTR lpCaption,UNIT uType)
```

四个参数分别是:

- hWnd: 父窗口的句柄;
- lpText: 要显示字符串的指针;
- lpCaption: 对话框标题字符串的指针;
- uType: 显示在对话框上的小图标类型。

使用这个函数要包含 windows.h 头文件。

调试一下,怎么样?

窗口上弹出了一个“第一个 Windows C 程序”对话框,上面有一行字:“Hello,World!”。

世界真的很美好啊!!

深入编程

□ 在 C 语言中,函数的声明如果没有指明返回类型,缺省值为 void,这个程序的主函数就没有返回

值。不过,在 Windows 编程时,我们最好养成个好习惯,指明函数的返回值类型,因为在 C++ 中,函数返回值类型是不可以缺省的。而我们在 Windows C 编程时,还是会用到 C++ 的一些概念,这样做,有利于以后深入地学习。

规范一点的程序应该是这样的:

```
#include<windows.h>
int APIENTRY winMain(HINSTANCE hInstance,HINSTANCE
hPrevInstance,
LPSTR lpCmdLine,int nCmdShow)
{
    MessageBox(NULL,"Hello,world!", "第一个windows C程序",
    MB_OK|MB_ICONASTERISK);
    return 0;
}
```

这里,我们声明的类型为 int 型,并且返回一个值 0,这样的函数就可以使用在复杂一点的函数调用中了。

□ 在这一小节中,我们有几处都提到了句柄的概念,句柄和指针的概念不同,它是作为操作系统内部索引表中的一个值来使用的,这样可以防止应用程序直接访问对象的内部结构,体现了 Windows 资源管理的优越性。譬如说,一个窗口打开之后,对应着内存中的一个内存块,这个窗口所在的内存块地址往往会由操作系统做动态的调整,但其句柄却不会随之变化。通过句柄就可以访问这个窗口,所以在使用的时候,可以把它当做指针一样看待。

二、获取本地计算机的主机名和 IP 地址

和 C 语言一样,函数是 Windows C 编程的最基本的单位。不过,Windows C 主要使用 API 函数,而网络编程则主要使用 Winsock 提供的 API 函数。

Winsock 是 90 年代初,为了方便网络编程由 Microsoft 联合了其他几家公司共同制定的一套 WINDOWS 下的网络编程接口,它是通过 C 语言的动态链接库方式提供给用户及软件开发者的,主要由 winsock.h 头文件和动态链接库 winsock.dll 组成,目前有两个版本:Winsock1.1 和 Winsock2.0。

在 Win32 平台上,访问众多的基层网络协议,Winsock 是首选接口。

用 Visual C++6.0 编译 Windows C 程序,使用 Winsock API 函数时,首先要把 wsock32.lib 添加到它的库模块中,否则在链接的时候会出现“error

LNK2001”错误。

添加 wsock32.lib 的具体步骤是:打开“工程”菜单,选择“设置”,在弹出的“Project settings”对话框中,点击“link”选项卡,然后在对象/库模块文本框中添加“wsock32.lib”。

最简单的网络编程是获取本机的主机名和 IP 地址,这个程序使用了 WSAStart()、WSACleanup()、gethostname()、gethostbyname() 四个 Winsock API 函数,这四个函数的功能和使用方法介绍如下:

1. WSAStartup():

【函数原型】

```
int PASCAL FAR WSAStartup(WORD
wVersionRequired, LPWSADATA lpWSADATA);
```

【使用说明】

每一个使用 winsock 的应用程序,都必须进行 WSAStart 函数调用,并且只有在调用成功之后才能使用其它的 Winsock 网络操作函数。

wVersionRequired: <输入参数>表示欲使用的 Winsock 版本,这是一个 WORD 类型的整数,它的高位字节定义的是次版本号,低位字节定义的是主版本号。

lpWSADATA: <输出参数>是一个指向 WSADATA 资料的指针。这个资料我们一般不使用。

返回值:调用成功返回 0;否则,返回出错信息。

2. WSACleanup():

【函数原型】

```
int PASCAL FAR WSACleanup(void);
```

【使用说明】

winsock 使用后,要调用 WSACleanup 函数关闭网络设备,以便释放其占用的资源。

3. gethostname()

【函数原型】

```
int PASCAL FAR gethostname(char FAR *
name, int namelen);
```

【使用说明】

该函数可以获取本地主机的主机名,其中:

name: <输出参数>用于指向所获取的主机名的缓冲区的指针。

Namelen: <输入参数>缓冲区的大小,以字节为

技术专区

Windows 开发

单位。

返回值：若无错误，返回 0；否则，返回错误代码。

4. gethostbyname()

【函数原型】

```
struct hostent FAR * PASCAL FAR
gethostbyname(const char FAR * name);
```

【使用说明】

该函数可以从主机名数据库中得到对应的“主机”。

该函数唯一的参数 `name` 就是前面调用函数 `gethostname()` 得到的主机名。若无错误，刚返回一个指向 `hostent` 结构的批针，它可以标识一个“主机”列表。

`Hostent` 结构定义如下：

```
Struct hostent
{
char FAR * h_name;
char FAR FAR ** h_aliases;
short h_addrtype;
char FAR FAR ** h_addr_list;
}
```

其中：

`h_name`：<输入参数>主机名地址（PC）。

`h_aliases`：一个由主机备用名组成的空终止数组。

`h_addrtype`：返回地址的类型，对于 Winsock，这个域总是 `PF_INET`。

`h_lenth`：每个地址的长度（字节数），对应于 `PF_INET` 域应该为 4。

`h_addr_list`：应该以空指针结尾的主机地址的列表，返回的地址是以网络顺序排列的。

其中，`h_addr_list[0]` 存放的就是本地主机的 4 个字节的 IP 地址，即：`h_addr_list[0][0].h_addr_list[0][1].h_addr_list[0][2].h_addr_list[0][3]`

一个简单的用消息框显示主机名和 IP 地址的源程序如下：

```
#include<winsock.h>

int WSA_return;
WSADATA WSAData;

HOSTENT *host_entry;
char host_name[256];
char host_address[256];

int APIENTRY WinMain(HINSTANCE hInstance,HINSTANCE
```

```
hPrevInstance, LPSTR lpCmdLine,int nCmdShow)
{
    WSA_return=WSAStartup(0x0101,&WSAData);

    if(WSA_return==0)
    {
        gethostname(host_name,256);

        host_entry=gethostbyname(host_name);

        if(host_entry!=0)
        {
            wsprintf(host_address,"%d.%d.%d.%d",
                (host_entry->h_addr_list[0][0]&0x00ff),
                (host_entry->h_addr_list[0][1]&0x00ff),
                (host_entry->h_addr_list[0][2]&0x00ff),
                (host_entry->h_addr_list[0][3]&0x00ff));

            MessageBox(NULL,host_address,host_name,MB_OK);
        }
    }

    WSACleanup();

    return 0;
}
```

深入编程

□ 前面显示 IP 地址的时候，我们使用的是消息框，规范一点的编程应该使用对话框，如何编辑一个对话框，很多书中都有介绍，编辑的对话框可参考图 5 的运行界面。

头文件 `Get_IP.h` 如下：

```
BOOL APIENTRY Hostname_ipDlgPro(HWND hDlg,UINT message,
WPARAM wParam,LPARAM lParam);
```

这个程序只使用了一个对话框过程，一般把这个过程的声明放在头文件中。

□ 源程序 `Get_IP.c`：

```
#include<winsock2.h>
#include"Get_IP.h"
#include"resource.h"
//这个头文件在创建资源的时候会自动生成，
//并会在插入资源时自动生成控件标识号。
int WSA_return;
WSADATA WSAData;

HOSTENT *host_entry;
char host_name[256];
char host_address[256];

int APIENTRY WinMain(HINSTANCE hInstance,HINSTANCE
hPrevInstance, LPSTR lpCmdLine,int nCmdShow)
{
    WSA_return=WSAStartup(0x0101,&WSAData);

    if(WSA_return==0)
    {
```

```

    gethostname(host_name,256);

    host_entry=gethostbyname(host_name);

    if(host_entry!=0)
    {
        wsprintf(host_address,"%d.%d.%d.%d",
            (host_entry->h_addr_list[0][0]&0x00ff),
            (host_entry->h_addr_list[0][1]&0x00ff),
            (host_entry->h_addr_list[0][2]&0x00ff),
            (host_entry->h_addr_list[0][3]&0x00ff));
    }

    WSACleanup();

    DialogBox(hInstance,"DIALOG1",NULL,(DLGPROC)
    Hostname_ipDlgPro);

    return 0;
}

BOOL APIENTRY Hostname_ipDlgPro(HWND hDlg,UINT message,
WPARAM wParam,LPARAM lParam)
{
    switch(message)
    {
        case WM_INITDIALOG:
            return(TRUE);
        case WM_COMMAND:
            if(LOWORD(wParam)==IDOK)
            {
                SetDlgItemText(hDlg,IDC_EDIT1,host_name);
                SetDlgItemText(hDlg,IDC_EDIT2,
                host_address);
                SetDlgItemText(hDlg,IDCANCEL,"确定");
            }

            if(LOWORD(wParam)==IDCANCEL)
                EndDialog(hDlg,TRUE);

            return(TRUE);
            break;
        }
        return(FALSE);
    }
}

```

三、利用 VisualC++6.0 编译 Windows C 程序

利用 Visual C++6.0 编译 Windows C 程序一般要经过以下四个步骤：新建项目、添加代码、添加资源和编译链接。下面我们简单地说明上面所介绍的获取本机的主机名和 IP 地址程序的编译过程：

一、新建项目

1. 启动 Microsoft Visual C++, 然后在【文件】菜单中先择【新建】命令, 弹出如图 1 所示的【新建】对话框：

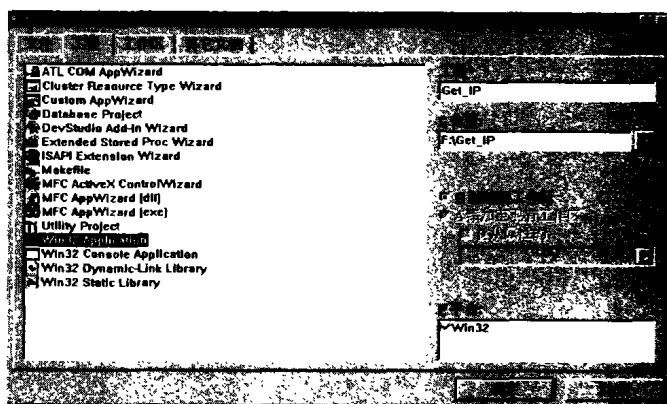


图 1

2. 在【新建】对话框中, 系统打开的是默认的【工程】选项卡, 【工程】选项卡左侧的列表框中有多种建立工程的方式, 我们选中“Win32 Application”选项。

3. 在【位置】文本框中输入新建工程的路径(例如: F:\), 在【工程】文本框中输入工程名称(例如: Get_IP)。

4. 选中【平台】列表框中的 Win32 复选框, 然后单击【确定】按钮。

5. 在随后的对话框中, 都选择默认设置, 完成后, 进入图 2 所示界面:

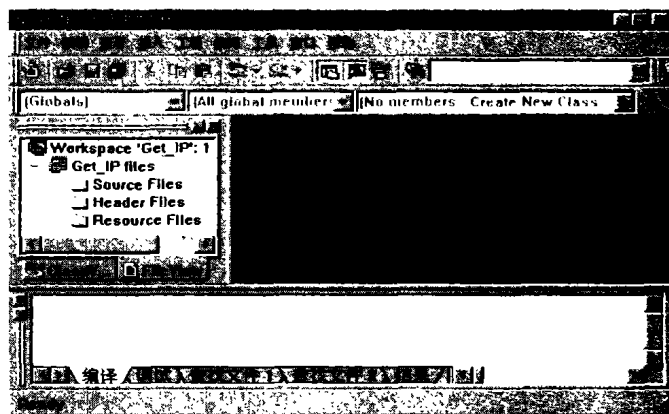


图 2

二、添加代码

在 Visual C++6.0 中, 源代码一般存放在源代码文件和头文件中, 往项目中添加源代码是非常方便的, 为项目新建一个源代码文件一般要按下述方法操作:

1. 选择【工程】|【添加工程】|【新建】选项, 弹出图 3 (见下页) 所示【新建】对话框。

2. 在对话框的【文件】选项卡中, 左侧的列表框选中“C++ Source File”选项, 右侧选中【添加工程】复选框, 并在【文件】文本框中输入源文件名(例如: Get_IP.c)。

技术专区

Windows 开发

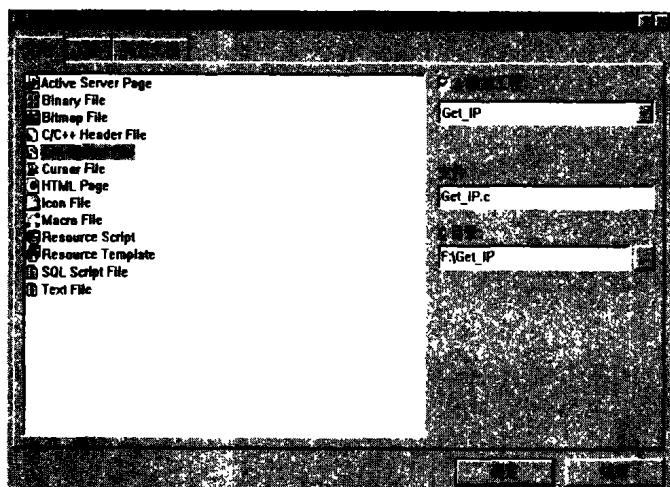


图 3

3. 单击【确定】按钮，【新建】对话框将被关闭，用户就可以在新建的 Get_IP.c 中输入程序的源代码了。

4. 添加头文件 Get_IP.h 的方法和上面所述过程一样，只是在【文件】选项卡中，左侧的列表框要选中“C/C++ Header File”选项。在【文件】文本框中输入头文件名（例如：Get_IP.h）。

三、添加资源

在添加资源前，必须在项目中先添加一个资源文件，然后可利用 Visual C++ 6.0 提供的资源编辑器为项目新建一个资源，具体步骤如下：

1. 选择【工程】|【添加工程】|【新建】选项，弹出图 3 所示【新建】对话框。

2. 在对话框的【文件】选项卡中，左侧的列表框选中“Resource Script”选项，右侧选中【添加工程】复选框，并在【文件】文本框中输入资源文件名（例如：Get_IP.rc）。

3. 单击确定，回到主窗口后，选择【插入】|【资源】选项，打开【插入资源】对话框，如图 4 所示，在【资源类型】列表框中选中“Dialog”选项，单击【新建】按钮，返回主窗口后，即可利用对话框编辑

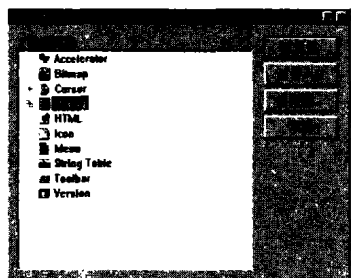


图 4

器进行编辑了。编辑后的对话框如图 5。

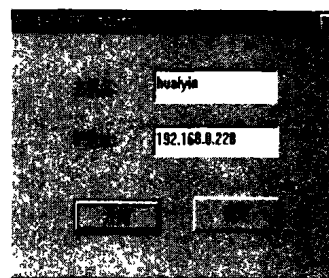


图 5

四、编译链接

在添加了源代码与资源文件后，就可以对程序编译链接了，可按 Ctrl+F7 键编译，按 F7 键链接，按 Ctrl+F5 键运行程序。在链接前要注意资源文件 Get_IP.rc 也要进行编译。

由于这个程序引用了 Winsock API 函数，在编译链接前，还要添加 wsock32.lib，具体方法前面已经介绍过，这里就不再赘述了。

一点看法

利用 C 语言编写 Windows 应用程序有两种方式：一种是 Windows C 编程方式，另一种是 Visual C++ 编程方式。在一般情况下，Visual C++ 编程方式编写的程序源代码量小、开发时的工作量小、工作难度也较小，但编译后的代码量较大，运行速度略低；而 Windows C 编程方式编写的程序源代码量虽然较大，但可执行代码效率高。随着技术的进步，Visual C++ 编程方式已被广泛采用，但象网络编程等一些对速度要求高、对硬件操作较多的程序，大多数还是用 Windows C 编程方式开发的。另外，学习 Windows C 程序设计，还有助于更深入地了解 Windows 的内幕和 Windows API。

从教学角度讲，在学生具备了 C 语言和其它一些前导课程基础后，直接进入 Windows C 网络编程等实用编程技术课程，不仅可以让学生尽早地接触到前沿的实用编程技术，而且还可以极大地调动学生的学习积极性，在有限的时间内，学到更多的知识和技术。C

注：本文讲述的编程实例都已编译通过，请到 CSDN 杂志频道 (<http://mag.csdn.net>) 下载。